A METHOD OF PROVIDING A CODE UPGRADE TO A HOST DEVICE HAVING A SMART CARD INTERFACE

BACKGROUND OF THE INVENTION

[0001] The present invention relates to a digital set-top box and, more particularly, to a method of upgrading the firmware/software of a digital set-top box using a standardized point of deployment (POD) interface.

[0002] With the advent of digital broadcast systems, a broad array of interactive entertainment services are now provided by multiple cable operators. Exemplary interactive services include, interactive program guides, email and web access, interactive game shows, video games and the like. Yet these interactive services must be interfaced through a consumer set-top box (i.e., a cable television converter). Traditionally, each manufacture of cable television (CATV) equipment had its own proprietary set-top box. This hindered computation, however, as a consumer who wanted to switch cable companies often needed to purchase a new set-top box. The Telecommunications Act of 1996 was enacted to provide an open cable hardware platform to provide interoperability and portability between set-top boxes and cable systems.

[0003] The Telecommunications Act of 1996 empowered the Federal Communications Commission (F.C.C.) to create regulations that would enable consumers the ability to obtain set-top boxes and similar equipment from sources other than cable operators. Prior to the act, cable equipment (i.e., decoder boxes) was not interoperable across systems, in other words, cable operators carved out geographical markets based on proprietary systems which required specific hardware/software for reception. As television content increasingly migrated from RF transmission schemes to cable/satellite based systems, consumers became reliant on their local cable operator for delivering the signal/interactive services to their home.

[0004] To accomplish the objective of the act, a universal set-top box hardware and software platform has been developed by North American cable operators known as the OpenCable® standard. In order to facilitate the

desired portability and interoperability (i.e., between cable head ends and set-top boxes), while maintaining the proprietary content of cable operators (i.e., encryption, security, pay programming) the OpenCable® compliant set-top boxes include generic hardware and software that may be used to decode cable television signals and a Point of Deployment (POD) interface for communicating with a POD module.

[0005] Although the invention is described in terms of an OpenCable® compliant set-top box, it is not limited to this standard. It may be practiced with any set-top box that is configured to accept a smart card that controls features of the set-top box including, without limitation, conditional access. In the materials that follow, the term open cable set-top box is used to refer to these devices generically.

[0006] The POD module includes hardware and software that customizes the set-top box to a particular CATV provider. As currently implemented, the POD interface to the set-top box is a personal computer memory card international association (PCMCIA) port, Japan electronic industry development association (JEIDA) port or the like, hereafter "smart card port." A compatible POD module (i.e. a PCMCIA or JEIDA card) interfaces with the smart card port to provide proprietary services (i.e., conditional access functions) such as pay programming, entertainment guides, encryption, security and the like. For the typical bi-directional cable system, the POD module includes a CPU channel, an out-band channel and an in-band channel operably linked to the smart card port of the set-top box. The set-top box relies on the POD module only for specific proprietary services. Because it includes generic hardware and software as well as the smart card port, the OpenCable® compliant set-top box is designed to be portable and interoperable across cable networks.

[0007] As firmware/software executables of these compliant set-top boxes are introduced for utilizing the standard, a method of upgrading, providing software patches, bug fixes and the like to the set-top box is necessary. Yet, the number of manufactures and anticipated set-top box models is too great to provide such upgrades over the cable systems even if cable operators were willing to dedicate the bandwidth. Moreover the size of

the executables would be prohibitive for provision via a telephone interface, similarly, attention by an authorized service provider is cost prohibitive.

[0008] Accordingly, there is a need for a method of providing a code upgrade to an open cable complaint host device in a cost effective manner.

SUMMARY OF THE INVENTION

[0009] The present invention provides a smart card which includes a software upgrade to a host device having a smart card interface. A memory in the smart card stores the software upgrade for delivery to the host device. The smart card includes a card information structure (CIS) in accordance with a standard protocol for identifying the smart card as a software upgrade card so that the host device can access the software upgrade.

[0010] The invention further provides a method of providing a code upgrade to a host device having a smart card interface. A smart card is provided including a code upgrade for transfer to the host device. The smart card is interfaced with the smart card interface of the host device. The smart card is recognized as a code upgrade and data on the card is accessed for upgrading the software/firmware of the host device.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] These and other features, aspects, and advantages of the present invention will become more fully apparent from the following description, appended claims, and accompanying drawings in which:

[0012] Fig. 1 is a block diagram of an open cable compliant host device having a POD interface and upgrade card in accordance with an exemplary embodiment of the invention;

[0013] Fig. 2 is a block diagram of the software interface provided for a smart card port of an exemplary host device in accordance with the present invention; and

[0014] Fig. 3 is a flow chart illustrating a method of upgrading code of the compliant host device in accordance with an exemplary embodiment of the present invention.

DESCRIPTION OF THE EXEMPLARY EMBODIMENT

[0015] Certain terminology used in the following description is for convenience only and is not limiting. The term "smart card" refers to portable circuit cards such as type I – III personal computer memory card international association (PCMCIA) cards, Japan electronic industry development association (JEIDA) cards and like cards configured to operate in accordance with the standardized mechanical and electrical specifications for use with open cable compliant host devices. The term "compliant host device" is defined as corresponding to the set-top box standard outlined in OpenCable® HOST-POD INTERFACE SPECIFICATION of Cable Television Laboratories Inc. the entirety of which is hereby incorporated by reference. The terms "code", "computer code" and "software" are used interchangeably. The term "tuple" is used to denote a data object having two or more components. In the drawings, the same reference numerals are used for designating the same elements throughout the several figures. While the invention is described in terms of an open cable compliant set-top-box, it is generally applicable to any set-top-box having a smart card interface.

[0016] The present invention is embodied in a smart card and smart card interface for providing a code upgrade to an open cable compliant host device. A memory of the smart card stores the code upgrade for delivery to the host device. The smart card having the code upgrade includes a card information structure (CIS) for identifying the smart card as a code upgrade card so that the complaint host device can access the code upgrade of the PCMCIA card to upgrade the software/firmware of the compliant host device.

[0017] The present invention is also embodied in a method of providing a cost effective code upgrade to an open cable complaint host device. The smart card storing the code upgrade is lightweight, small in dimension and relatively inexpensive to produce. Thus, the smart card in accordance with

the present invention can be readily supplied to an operator via normal business distribution channels such as mail services for correcting software errors in, or for upgrading an open cable complaint host device.

[0018] I. Conventional System Components

[0019] Referring now to the drawings, Fig. 1 is a high level block diagram showing a typical open cable compliant system 5. The system 5 is shown in a typical configuration, having an open cable compliant host device 7 operably linked to a point of deployment (POD) device 10 via a POD interface 42 and a subscriber line 11 linked to a cable television (CATV) head end. The POD 10 is provided to each subscriber of a proprietary cable system. The POD 10 enables the isolation of cable operator hardware/software specifics from the host device 7 (i.e., set-top box). In this way, the POD 10 and host device 7 split signaling functions such that RF and QPSK processing are performed by the host device 10 and decryption, descrambling and any signal processing specific to the CATV provider is performed under control of the POD 10. The POD 10 also provides an interface for communications between the subscriber and the CATV head end.

[0020] While the exemplary embodiment is described with reference to a bi-directional (i.e., two-way) land line cable system, those skilled in the art will recognize that other types of cable systems, including wireless systems are operable under the open cable standard. Similarly, one-way systems, and data over cable service interface specification (DOCSIS) systems are likewise compatible with the teachings of the present invention.

[0021] In the exemplary system, the host device 7 includes tuner 12, demodulator 14, quadrature phase shift keying (QPSK) transmitter 16, receiver 18, data processor 22 de-multiplexer 20, POD interface 42, random access memory (RAM) 23 and optional read only memory (ROM) 35. Both of the memories 23 and 35 are coupled to the data processor 22. The random access memory may be, for example, a battery backed dynamic RAM or other nonvolatile memory. As described below, the optional ROM 35 may hold a bootstrap loader and safe versions of the operational software. The

exemplary processor 22 may run the software directly from the ROM 35 or download it to the RAM 23 if an error is detected in the current operating software stored in the RAM 23.

[0022] In operation, the transmitter 16 and receiver 18 of the host device 7 are used to transfer data between the set-top box and the CATV head-end via the cable or subscriber line 11. The tuner 12 of the host device 7 is provided for tuning to a desired channel frequency for receiving A/V content from the subscriber line 11. The demodulator 14 of the exemplary system 5 is a quadrature amplitude modulation QAM demodulator. The demodulated content is provided to the POD 10 for granting access to the subscriber.

[0023] The POD interface 42 is a 68 pin smart card slot for receiving a POD 10 or smart card 25 (as shown in Fig. 2). The POD interface may operate in 16-bit or 32 bit mode as necessary.

[0024] The POD device 10, in accordance with an instruction set and subscriber permission parameters, conditionally grants/denies access to content provided via the subscriber line. Where the subscriber is granted access, POD 10 may provide a conditional access key, or "descrambling means" via the CPU channel to data processor 22 for presenting the content to the viewer via host device 7.

[0025] For example, the CATV signal may include a channel that carries an encrypted digital television signal. This television signal may be encoded, for example, according to the moving pictures experts group (MPEG) standard and encrypted using a symmetric algorithm such as the triple data encryption standard (triple DES). In one example, the packetized elementary stream (PES) packets of the MPEG bit-stream may be encrypted and then packed into MPEG transport packets. The transport decoder of the set-top box may receive and demodulate the signal, recover the transport packets and reconstruct the encrypted PES packets. The data in these packets is then passed to the in-band channel of the POD module where it is decrypted and passed back to the transport decoder. The POD module may use a pre-stored decryption key that may be provided, for example, via the

out-of-band channel through an earlier communication with the cable head-end in which the encrypted programming was ordered.

[0026] The de-multiplexer 20 of host device 7 "unpacks" data packets of the subscriber line. The de-multiplexer may, for example, separate the data from PES framing to recover an elementary bit-stream. Alternatively, if the transport packets are encrypted, the de-multiplexer 20 may recover the PES packets from the transport packets and provide either the PES packets or the elementary bit-stream to an MPEG decoder (not shown).

[0027] Of course, the subject invention is not limited to any encoding standard nor to a digital implementation. For example, the POD module may be used to signal an analog decoder that the subscriber has paid for the programming that is being received and thus enable the decoder to apply conventional analog descrambling techniques to the received analog video signals.

[0028] The exemplary POD 10 of Fig. 1 is configured to operate on a bi-directional cable network. The POD 10 includes an out-of-band channel designated OOB, an in-band channel designated INB and a processing channel designated CPU. As described above, the INB signal receives content from the cable head end, the OOB channel allows two-way data communication between the set-top box and the head-end.

[0029] The content received via the INB channel is provided via tuner 12 and demodulator 14 of the host device 7. Where portions of the content are subject to conditional access (e.g., scrambled or encrypted) the POD 10 may provide a decryption facility including a conditional access key, or authorization for descrambling to present the content to the viewer via host device 7.

[0030] The OOB channel of the exemplary system 5 is provided for transmitting content such as a subscriber request to the cable head end for receiving a corresponding service or reply via the subscriber line. The QPSK transmitter of host device 7 is provided for modulating a carrier signal

transmitted through the OOB channel to output a data stream to the head-end via the subscriber line.

[0031] Upon connection of POD 10 or a smart card 25 (see Fig. 2) to POD interface 42 of host device 7, an initialization sequence in accordance with a standard protocol of smart card 25 is performed for enabling Plug and Play® functionality. The initialization sequence reads a card information structure (CIS) or "Metaformat" of a smart card 25 to determine whether the card is a POD. The Metaformat is a hierarchy of protocol layers and set of device independent drivers for specifying data-recording formats and organization. By programming the data processor 22 of host device 7 to recognize a tuple of the basic compatibility or "physical layer" of the smart card protocol, the host device can recognize a smart card 25 as a software/firmware upgrade and operate accordingly.

[0032] II. Smart card Protocol Layers

[0033] Referring now more specifically to Fig. 2, a block diagram of the host device software/firmware and Metaformat structure of the exemplary embodiment is shown.

[0034] The host device 7 includes application software 30, operating system 32, basic input output system (BIOS) 34, compatibility layer or "physical" layer 36, data recording/organization layer or "card services" 38, system specific layer or "socket services" 40, and POD interface 42. The system BIOS 34 is preloaded to the host device 7 upon manufacture (i.e., firmware). The application software 30 includes programs and data stored in the memory 23 (shown In Fig. 1) of host device 7 such as channel mapping, processing procedures, instruction sets, on-screen display bitmaps and the like. In addition to the memory 23, the host device 7 may also include a read-only memory (ROM) 35 which holds a copy of the operating system and BIOS as provided by the manufacturer. As a part of an error recovery procedure, following, for example, an unsuccessful software upgrade attempt, the host device 7 may reload the base software from the ROM 35.

[0035] The exemplary host device 7 employs BIOS 34 for communicating with peripheral devices in association with an operating system 32. As known to those in the art, the operating system 32 allocates the use of hardware resources of host device 7 for use in operation and or communication between hardware components. In the exemplary embodiment, the software upgrade may change the operating system 32, the BIOS 34 or both.

[0036] The Metaformat of the POD interface 42 is illustrated by compatibility layer 36, data recording/organization layer 38 and system specific layer 40. In accordance with the Metaformat, the smart card 25 can perform a plurality of memory and I/O operations such as providing data to the host device 7, providing additional memory capacity, and providing fax/modem capabilities, etc. The Metaformat is a standard protocol to integrate the smart card 25 into the host device 7 via POD interface 42.

[0037] System specific layer 40 of the Metaformat is the software interface to the POD interface 42 for managing all POD interfaces 42 and associated registers of a system 5. System specific layer 40 may be included in the host device 7 as a device driver and/or as a function of BIOS 34.

[0038] Data recording/organization layer 38 is an application programming interface (API) for enabling multiple applications 30 to communicate with smart card 25.

[0039] The compatibility layer or "physical layer" 36 specifies a minimum level of card data organization. Data at the card device driver layer include supported configurations, manufacturer, and individual device characteristics, etc.

[0040] In accordance with the OpenCable® specification, the CIS of the basic compatibility layer 40 enables the interconnection of a POD 10 for use with a compliant host device 7. Upon connecting the POD 10 or smart card 25 to the host device 7, the host device 7 reads the CIS to identify the smart card 25 or POD 10. The host device 7 may also identify the POD 10 with a specific application, for example, a conditional access protocol such as EIA-

layer. At step 306, the data processor 22 determines whether the smart card is a POD 10. If the smart card is a POD 10, at step 308 the POD 10 is initialized. If the smart card 25 is recognized as a software upgrade, the process proceeds to step 310, otherwise an error message is produced by host device 7. At step 310 the new software is read from the memory of smart card 25. At step 314, the data processor 22 monitors the transfer until completion. Where the transfer cannot be completed an error is displayed at step 316 and, at step 318, the process determines whether the software update should be attempted again. If the update is to be attempted again, control transfers to step 310, described above. If the update is not to be attempted again then, at step 320, the temporary memory is cleared and the process ends. If, at step 314, the process determines that the transfer was performed correctly, at step 311, the code from the smart card is copied to permanent memory overwriting the operational software and, at step 312, the host device is rebooted.

[0046] The subject invention may perform several steps to ensure that the software upgrade is proper for the particular set-top-box and that the upgrade is properly copied before the host device is rebooted to use the upgraded software. First, upon detecting that the smart card 25 contains a software upgrade, the host device 7 may compare data contained in the card 25 to ensure that the upgrade is compatible with the host device. This may include, for example, comparing manufacturer and model number data as well as comparing for compatibility with the current operating system 32 and BIOS 34 (shown in Fig. 2).

[0047] If the system determines that the upgrade is compatible, it transfers control to a bootstrap loader that performs the actual transfer. The bootstrap loader may, for example, calculate a checksum or cyclic redundancy code (CRC) as it loads the new operating software and compare the calculated value with a corresponding value contained in the smart card 25 upon completion of the process. If the two values do not match, the operator may be prompted by a message displayed on the display device or on a LCD screen of the host device 7 to retry the software upgrade or to return to the original software, which the bootstrap loader loads from the ROM 35, shown in Fig. 2. If the checksum or CRC values do match, the software

upgrade is permanently written into the memory 23. Upon successful completion of the update, the bootstrap loader, or the updated software may send a message to the head end via the OOB channel indicating that the software has been upgraded. This may be desirable, for example, to inform the head-end of new features that are enabled by the software upgrade. The bootstrap loader or base software from the ROM 35 may also indicate an unsuccessful upgrade by sending a message to the head end via the OOB channel.

[0048] Alternatively, as described above, the upgraded software code may be provided as a part of an applications program contained in the smart card 25. In this alternative embodiment of the invention, the software upgrade may be provided by the smart card 25 or the POD module 10. When provided via the POD module, the upgraded software may be included, for example, in applications data compliant with the national renewable security standard (NRSS). Details on NRSS may be found, for example, in a publication entitled "Conditional Access System for Terrestrial Broadcast" Doc. A/70 published by the Advanced Television Systems Committee (www.atsc.org). If this upgrade method is used, the current operational software running on the host system 7 may check the version and hardware compatibility of the new software, copy the new operational software to an area of memory, perform a checksum or CRC to ensure that the software is properly copied and then invoke the bootstrap loader to overwrite the current operational software in the memory 23.

[0049] It will be understood that various changes in the details, materials, and arrangements of the parts which have been described and illustrated above in order to explain the nature of this invention may be made by those skilled in the art without departing from the principle and scope of the invention as recited in the following claims.